Amendments to the Claims

1. (Currently Amended) A computer-implemented method comprising:

receiving a plurality of events;

applying the plurality of events to a correlation function, wherein the correlation function is implemented as a state machine in a first programming language and is 1) configured to correlate the plurality of events and 2) implemented using a schema which defines state classes and permits the use of a variety of different programming languages by developers;

identifying an event to which an update consumer has subscribed, wherein the update consumer is:

a class object separate from the state machine that defines transition operations for the state machine in said first programming language in lieu of when the state machine is defined; and

configured to update the state machine when the event to which the update consumer has subscribed occurs by invoking said transition operations;

applying the update consumer to the state machine in response to the identified event; and

generating a specific event if the correlation function is satisfied by the plurality of events.

2. (Canceled)

3. (Original) A method as recited in claim 1 further including:

receiving a data element; and

applying the data element and at least one of the plurality of events to the correlation function.

4. (Original) A method as recited in claim 1 further including:

receiving a plurality of data elements; and

applying the plurality of data elements and the plurality of events to the correlation function.

5. (Original) A method as recited in claim 1 further including communicating the specific event to at least one event consumer that subscribed to the specific event.

6. (Original) A method as recited in claim 1 further including continuing to receive additional events and apply the additional events to the correlation function if the correlation function is not satisfied by the plurality of events.

7. (Original) A method as recited in claim 1 further including resetting the correlation function after generating a specific event.

8. (Original) A method as recited in claim 1 further including:

creating an instance of a particular state machine; and

defining transitions for the particular state machine by subscribing to at least one event.

9. (Previously Presented) A method as recited in claim 8 further including deleting the instance of the particular state machine if the instance of the particular state machine reaches a final state.

10. (Currently Amended) One or more computer~~-readable memories~~ storage media containing a computer program that is ~~executable~~ executed by a processor to perform the method recited in claim 1.

11. (Currently Amended) A computer-implemented method comprising:

receiving a plurality of events;

receiving a plurality of data elements;

identifying a plurality of correlation functions configured to correlate the plurality of events and the plurality of data elements, wherein each correlation function is implemented with an associated state machine in a first programming language, and implemented using a schema which defines state classes and permits the use of a variety of different programming languages by developers; and wherein each state machine has an associated update consumer provided as a class object separate from the associated state machine that defines transition operations for the state machine in said first programming language in lieu of when the state machine is defined and that updates the state of the associated state machine based on an subscribed event;

applying the plurality of events and plurality of data elements to the plurality of correlation functions, including updating the state machine when the event to which the update consumer has subscribed occurs by invoking the transition operations; and

generating a specific event if at least one of the plurality of correlation functions is satisfied.

12. (Previously Presented) A method as recited in claim 11 further comprising deleting a particular state machine when the particular state machine reaches a final state.

13. (Canceled)

14. (Original) A method as recited in claim 11 further including communicating the specific event to at least one event consumer that subscribed to receive the specific event.

15. (Original) A method as recited in claim 11 further including:

receiving additional events;

receiving additional data elements; and

applying the plurality of events, the additional events, the plurality of data elements and the additional data elements to the plurality of correlation functions.

16. (Previously Presented) A method as recited in claim 11 further including:

receiving additional events;

receiving additional data elements;

receiving additional correlation functions; and

applying the plurality of events, the additional events, the plurality of data elements and the additional data elements to the plurality of correlation functions and the additional correlation functions.

17. (Original) A method as recited in claim 16 further including generating the specific event if at least one of the plurality of correlation functions or at least one of the additional correlation functions is satisfied.

18. (Original) A method as recited in claim 11 wherein the specific event generated is dependent on which correlation function is satisfied.

19. (Currently Amended) One or more computer~~-readable memories~~ storage media containing a computer program that is ~~executable~~ executed by a processor to perform the method recited in claim 11.

20. (Currently Amended) A computer-implemented method comprising:

identifying a schema for creating state machines, wherein the state machines ~~to~~ correlate at least two events and the schema defines state classes and permits the use of a variety of different programming languages by developers;

creating an instance of a particular state machine implemented in a first programming language;

defining transitions for the particular state machine in the first programming language by subscribing to at least one event by an update consumer; and

applying ~~an~~ the update consumer to the particular state machine to update the state of the particular state machine, wherein the update consumer is a class object provided separate from the particular state machine that defines and invokes transition operations for the state machine in lieu of when the state machine is defined.

21. (Original) A method as recited in claim 20 further including deleting the particular state machine if the particular state machine reaches a final state.

22. (Original) A method as recited in claim 20 wherein the particular state machine includes a timer, the method further including deleting the particular state machine if the timer expires.

23. (Original) A method as recited in claim 20 wherein the particular state machine correlates at least one event and at least one data element.

24. (Original) A method as recited in claim 20 wherein the particular state machine correlates a plurality of events and a plurality of data elements.

25. (Original) A method as recited in claim 20 further including determining a current state of the particular state machine.

26. (Canceled).

27. (Currently Amended) One or more computer~~-readable memories~~ storage media containing a computer program that is ~~executable~~ executed by a processor to perform the method recited in claim 20.

28. (Currently Amended) An apparatus comprising:

a processor;

a plurality of event consumers; and

an event correlator coupled to the plurality of event consumers, the event correlator ~~executable~~ executed via the processor to receive events from at least one event source and to receive data elements from at least one data source, the event correlator further to receive at least one correlation function configured to correlate events and data elements and to apply the received events and the received data elements to the correlation function, wherein the correlation function is implemented by a state machine in a first programming language using a schema which defines state classes and permits the use of a variety of different programming languages by developers ~~having~~ and the state machine has an associated update consumer provided as a class object separate from the state machine that defines and invokes transition operations for the state machine in lieu of when the state machine is defined, the transition operations ~~that~~ update the state of the state machine based on an subscribed event, and wherein the event correlator generates a specific event if the received events and the received data satisfy the correlation function including updating the state machine when the event to which the update consumer has subscribed occurs.

29. (Original) An apparatus as recited in claim 28 wherein the event correlator communicates the specific event to the plurality of event consumers.

30. (Original) An apparatus as recited in claim 28 wherein the event correlator communicates the specific event to event consumers that have requested to receive the specific event.

31. (Original) An apparatus as recited in claim 28 wherein the event correlator communicates the specific event to a plurality of filters, wherein each of the plurality of filters is associated with one of the plurality of event consumers.

32-33. (Canceled).

34. (Original) An apparatus as recited in claim 28 wherein the event correlator continues to receive additional events and additional data elements and apply the additional events and the additional data elements to the correlation function.

35. (Currently Amended) One or more computer-~~readable~~ storage media having stored thereon a computer program ~~that, when~~ executed by one or more processors, ~~causes~~ causing the one or more processors to:

receive a plurality of events;

identify a plurality of correlation functions configured to correlate the plurality of events, wherein each of the plurality of correlation functions is implemented as a state machine in a first programming language using a schema which defines state classes and permits the use of a variety of different programming languages by developers ~~having~~ and has an associated update consumer provided separate from the state machine that defines transition operations in said first programming language for the state machine, in lieu of the when the state machine is defined, to update the state of the state machine based on an subscribed event;

apply the plurality of events to the plurality of correlation functions to determine whether any of the plurality of correlation functions are satisfied by the plurality of events, wherein the plurality of events are applied by causing update consumers associated with each event to update the state of the associated state machine by invoking the transition operations when the associated subscribed event occurs; and

generate a specific event if one of the plurality of correlation functions is satisfied by the plurality of events.

36. (Canceled).

37. (Currently Amended) One or more computer-~~readable~~ storage media as recited in claim 35 wherein each state machine is a class object.

38. (Currently Amended) One or more computer computer-~~readable~~ storage media as recited in claim 37 further causing the one or more processors to identify a current state of the state machine.

39. (Currently Amended) One or more computer computer-~~readable~~ storage media as recited in claim 35 further causing the one or more processors to:

create a new instance of a state machine to implement a particular correlation function; and

define transitions for the new instance of the state machine by subscribing to at least one event.

40. (Currently Amended) A computer-implemented method comprising:

receiving events from event providers;

creating a first state machine in a first programming language using a schema which defines state classes and permits the use of a variety of different programming languages by developers;

creating a second state machine in a second programming language using a schema which defines state classes and permits the use of a variety of different programming languages by developers;

associating a first event type with the first state machine, wherein the first state machine has an associated first update consumer separate from the first state machine that defines first transition operations in a first programming language for the first state machine in lieu when the first state machine is defined, wherein the first transition operations update the state of the first state machine based on an subscribed event;

associating a second event type with the second state machine, wherein the second state machine has an associated second update consumer, separate from the second state machine, that defines second transition operations in a second programming language for the second state machine in lieu of when the second state machine is defined, wherein the second transition objects update the state of the second state machine based on an subscribed event;

in response to receiving an event having a first event type, applying the first update consumer to the first state machine by invoking the first transition operations;

in response to receiving an event having a second event type, applying the second update consumer to the second state machine by invoking the second transition operations; and

if the events are correlated:

generating an additional event; and

sending the additional event to the event consumer.

41. (Previously Presented) The method as recited in claim 40, further comprising deleting the first state machine if the first state machine reaches a final state.

42. (Previously Presented) The method as recited in claim 40, wherein the additional event is sent to the event consumer through a filter associated with the event consumer.

43. (Previously Presented) The method as recited in claim 40, wherein the event consumer includes at least one of an event logging consumer, an event forwarding consumer, a mail consumer, and a scripting consumer.